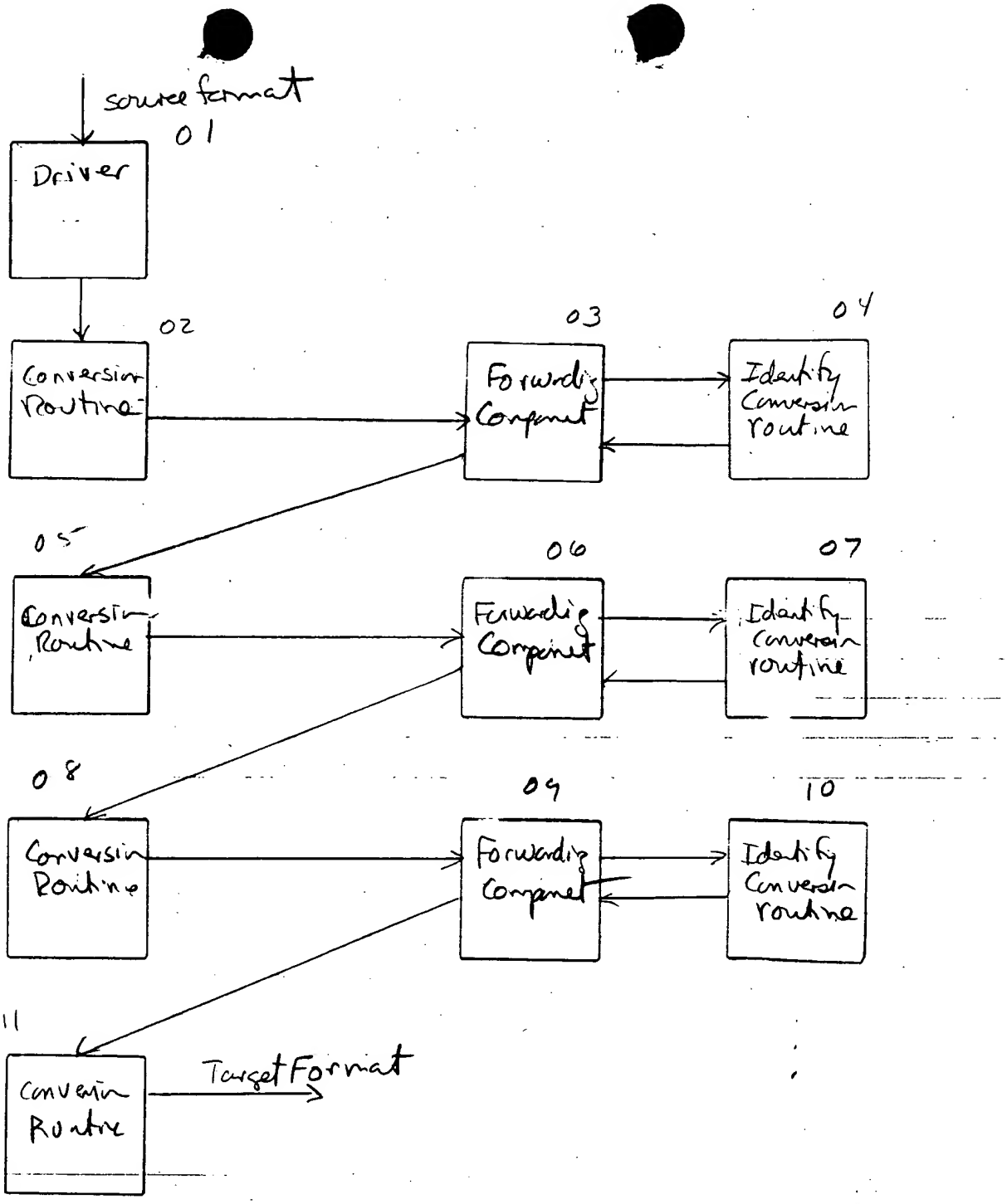

source format
01
Driver
02
Conversion Routine
03
Forwarding Component
04
Identify Conversion routine
05
Conversion Routine
06
Forwarding Component
07
Identify Conversion routine
08
Conversion Routine
09
Forwarding Component
10
Identify Conversion routine
11
Conversion Routine
Target Format

Fig 1

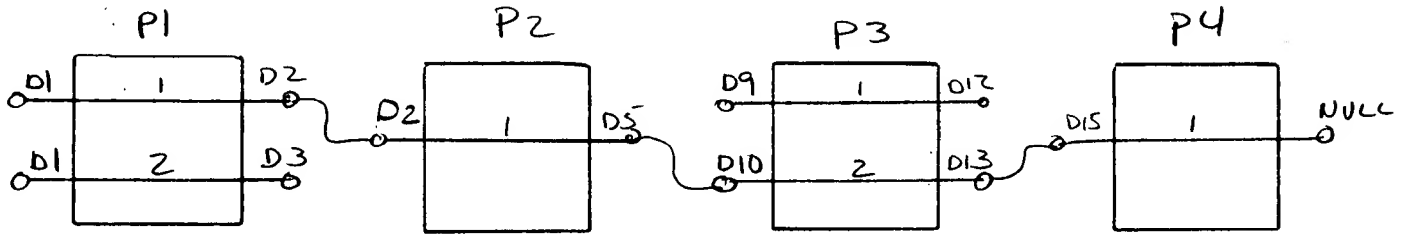


Fig 2

SEARCH REPORT

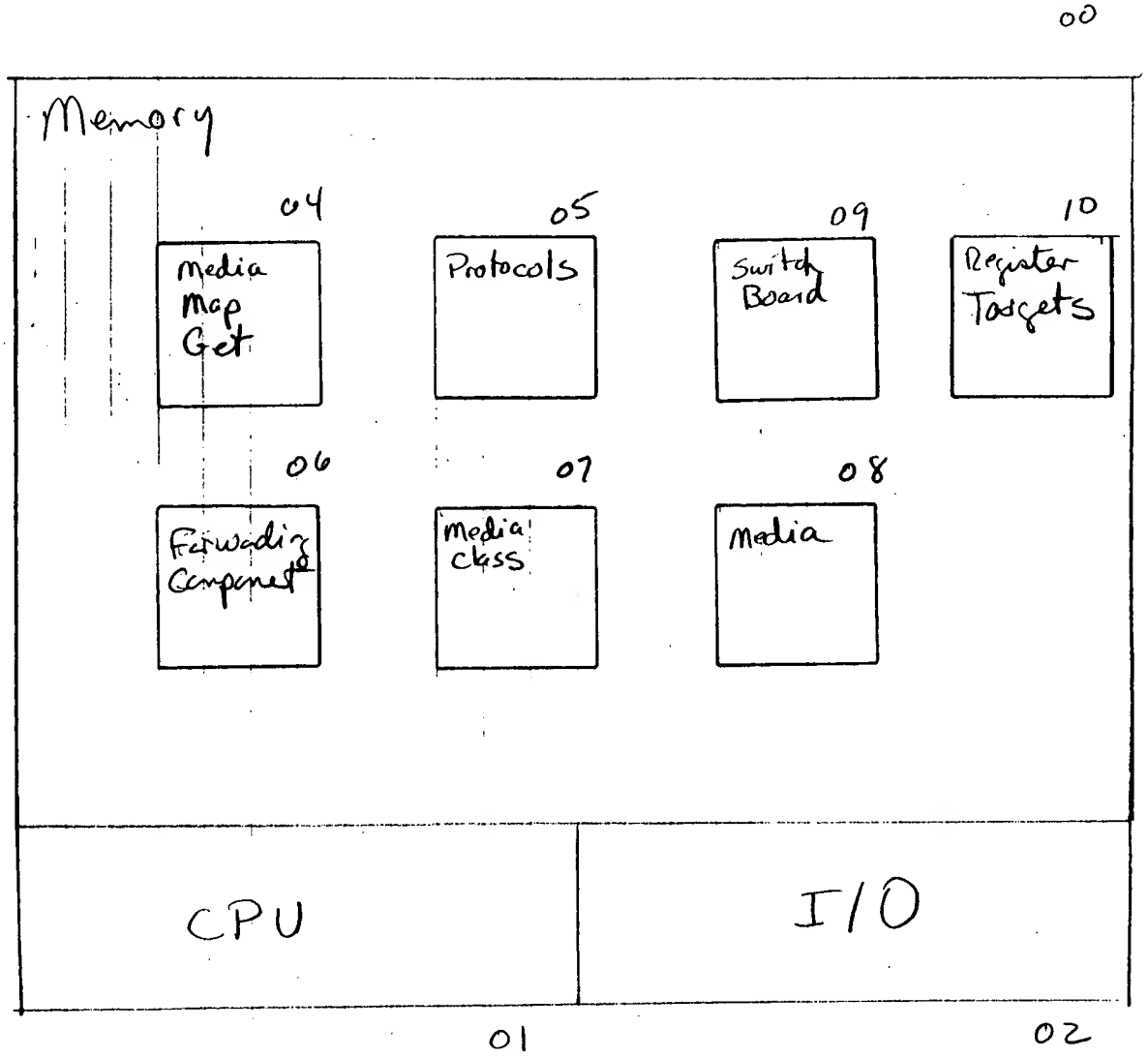


Fig 3

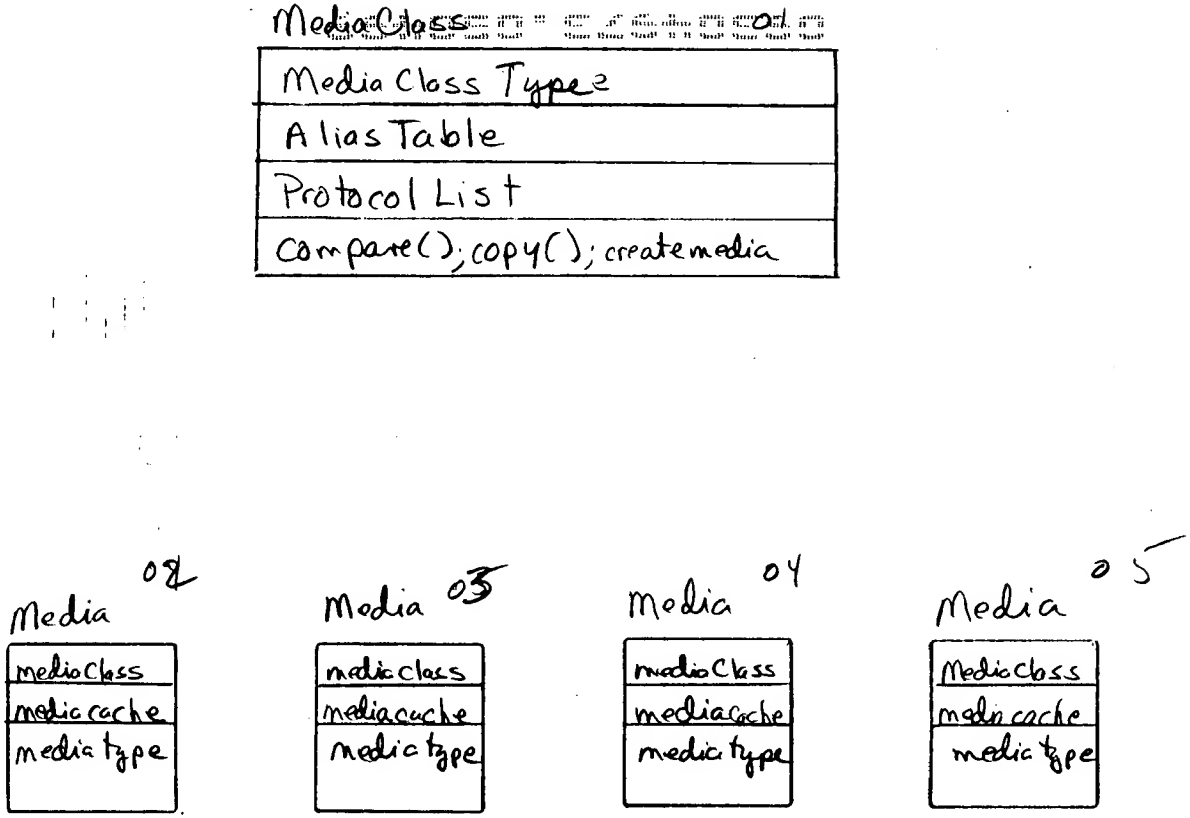

MediaClass
Media Class Type
Alias Table
Protocol List
compare(); copy(); create media
Media 02
mediaClass
mediacache
media type
Media 03
mediaclass
mediacache
media type
Media 04
mediaClass
mediacache
media type
Media 05
Mediaclass
mediacache
media type

Figure 4

Alias table

registered media	reference to registered media

Protocol List 02

pointer to protocol

MediaClass

Media

Media Cache 03

ref to registered target media	Address list

Figure 5

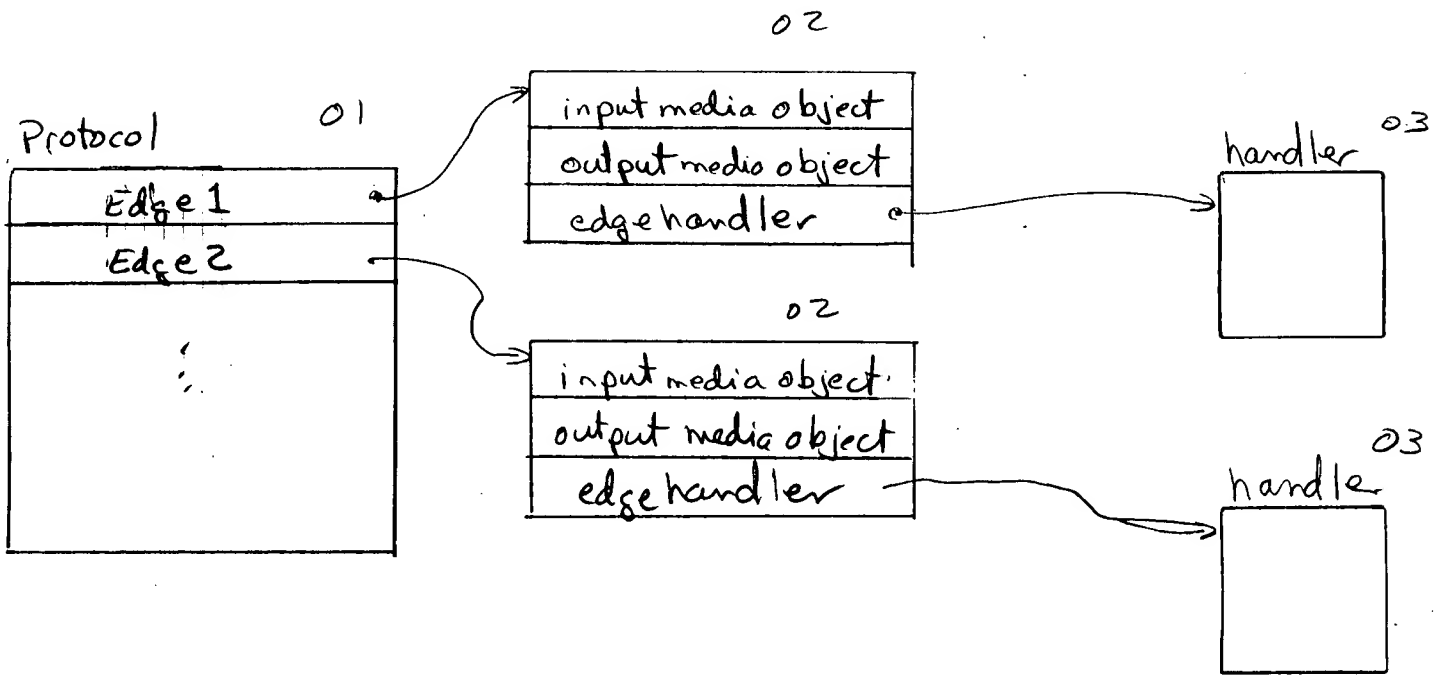

02
Protocol
01
Edge 1
Edge 2
input media object
output media object
edge handler
02
input media object
output media object
edge handler
handler
03
handler
03

Figure 6

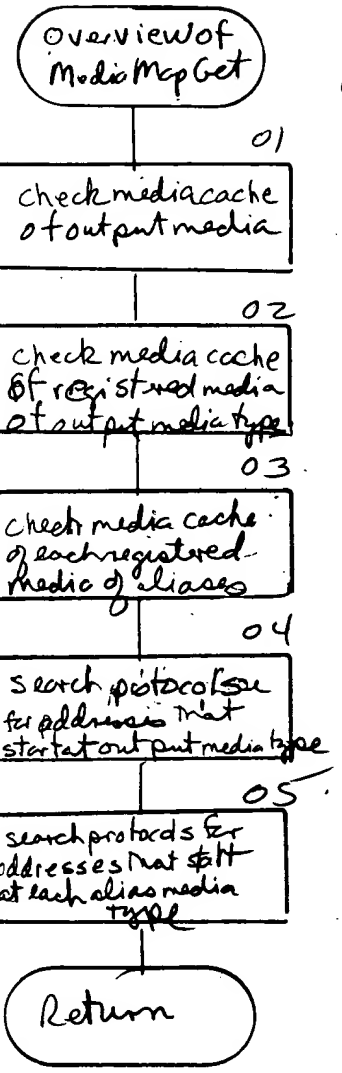

Overview of MediaMapGet
(output media, target media, address list)
01
check media cache of output media
02
check media cache of registered media of output media type
03
check media cache of each registered media of aliases
04
search protocols for addresses that start at output media type
05
search protocols for addresses that start at each aliases media type
Return

Figure 7

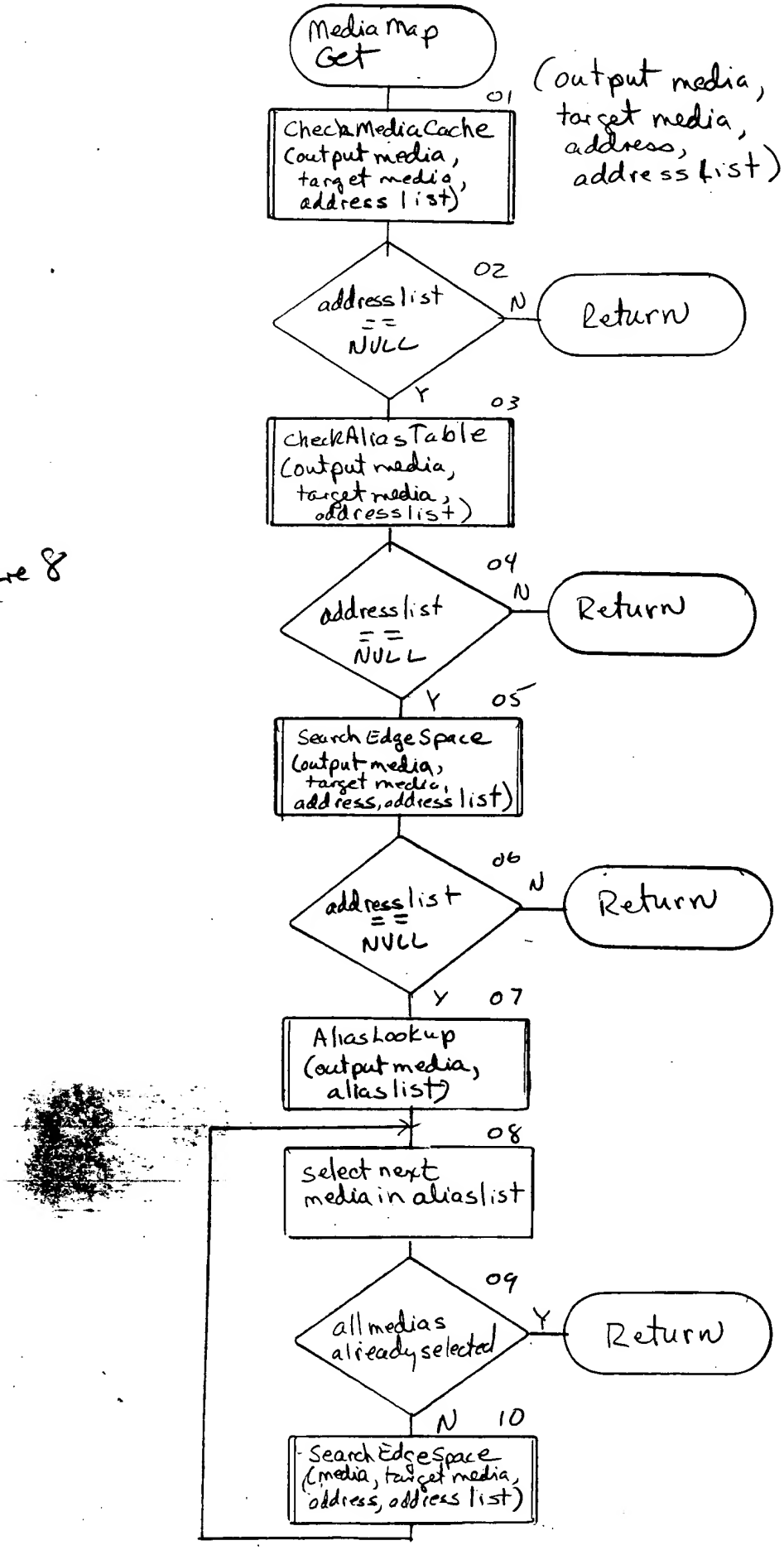

Media Map Get
(output media, target media, address, address list)
01
CheckMediaCache (output media, target media, address list)
02
address list == NULL
N
Return
Y
03
CheckAliasTable (output media, target media, address list)
04
address list == NULL
N
Return
Y
05
Search Edge Space (output media, target media, address, address list)
06
address list == NULL
N
Return
Y
07
AliasLookup (output media, alias list)
08
select next media in alias list
09
all medias already selected
Y
Return
N
10
Search Edge Space (media, target media, address, address list)

Figure 8

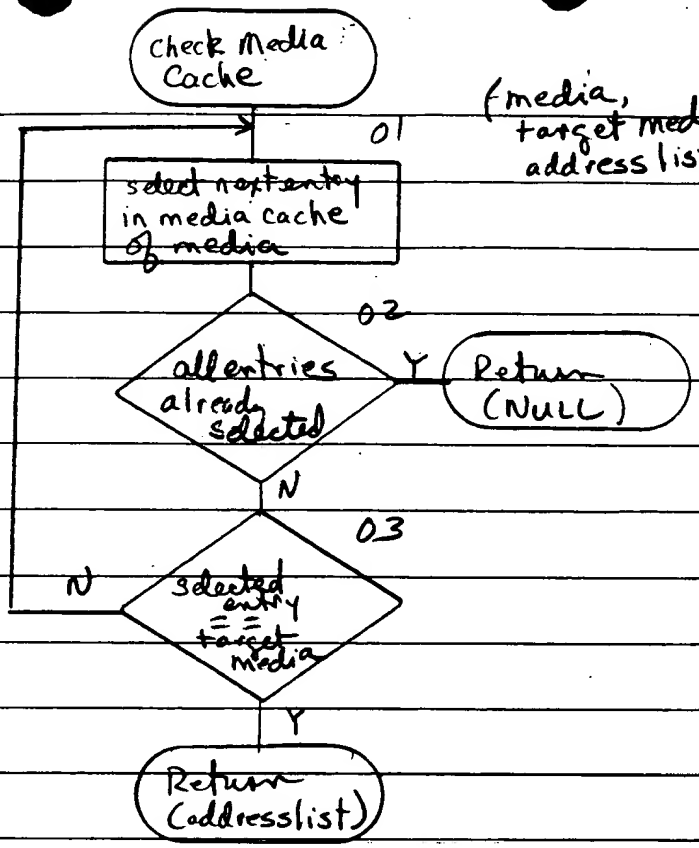

Check Media Cache
(media, target media, address list)
01
select next entry in media cache of media
02
all entries already selected
Y
Return (NULL)
N
03
selected entry == target media
N
Y
Return (addresslist)

Figure 9

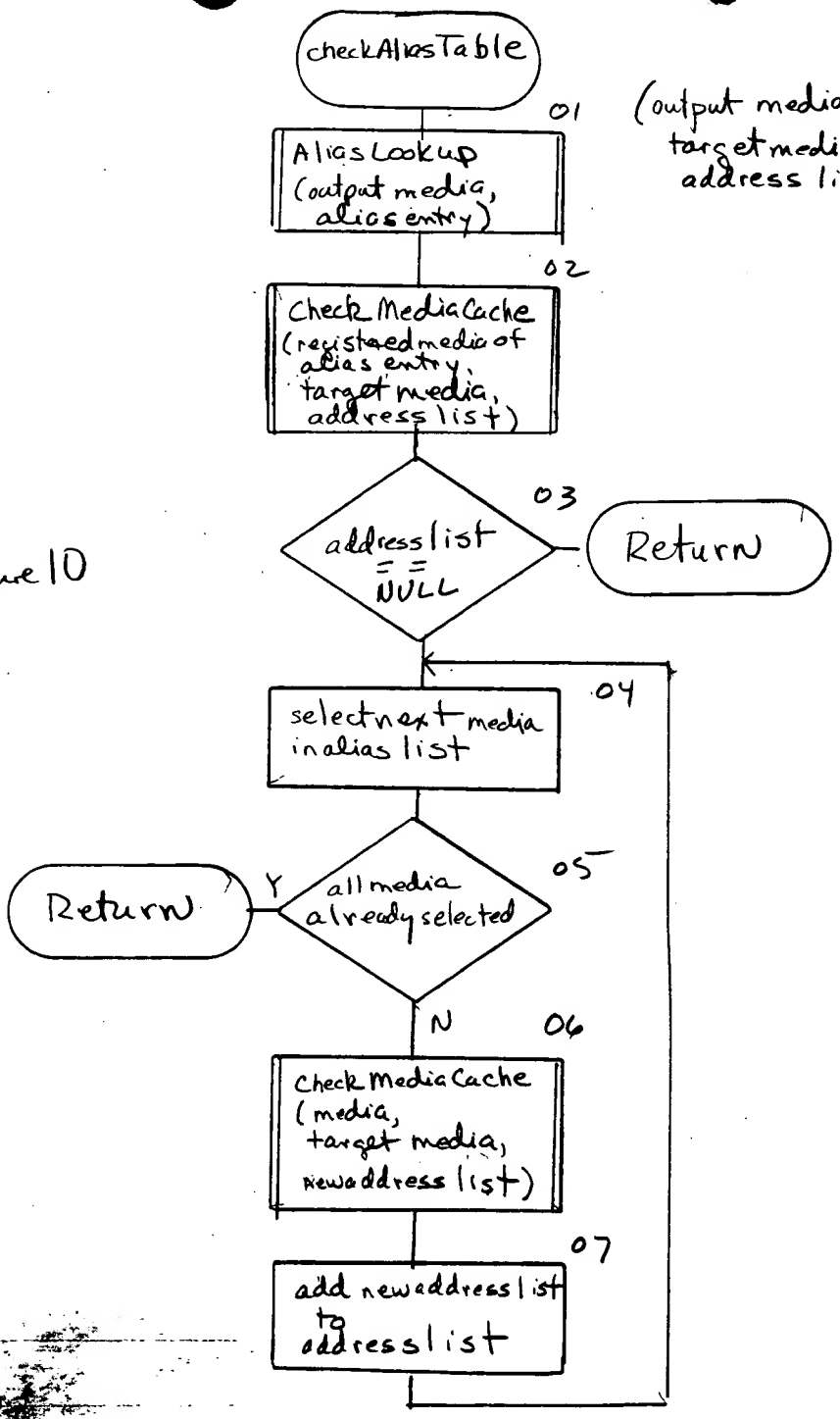

checkAliasTable
(output media, target media, address list)
01
AliasLookup (output media, alias entry)
02
CheckMediaCache (registered media of alias entry, target media, address list)
03
address list == NULL
Return
04
select next media in alias list
05
all media already selected
Y
Return
N
06
CheckMediaCache (media, target media, new address list)
07
add new address list to address list

Figure 10

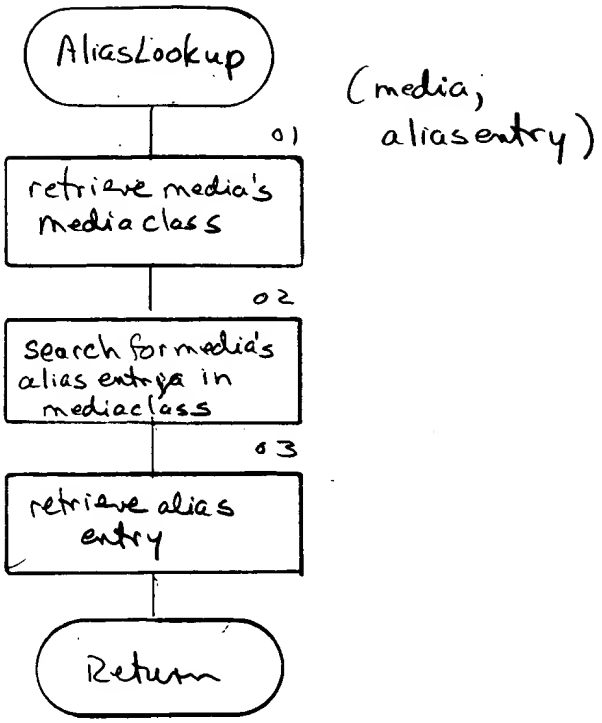

AliasLookup
(media, aliasentry)
01
retrieve media's mediaclass
02
search for media's alias entrys in mediaclass
03
retrieve alias entry
Return

Figure 11

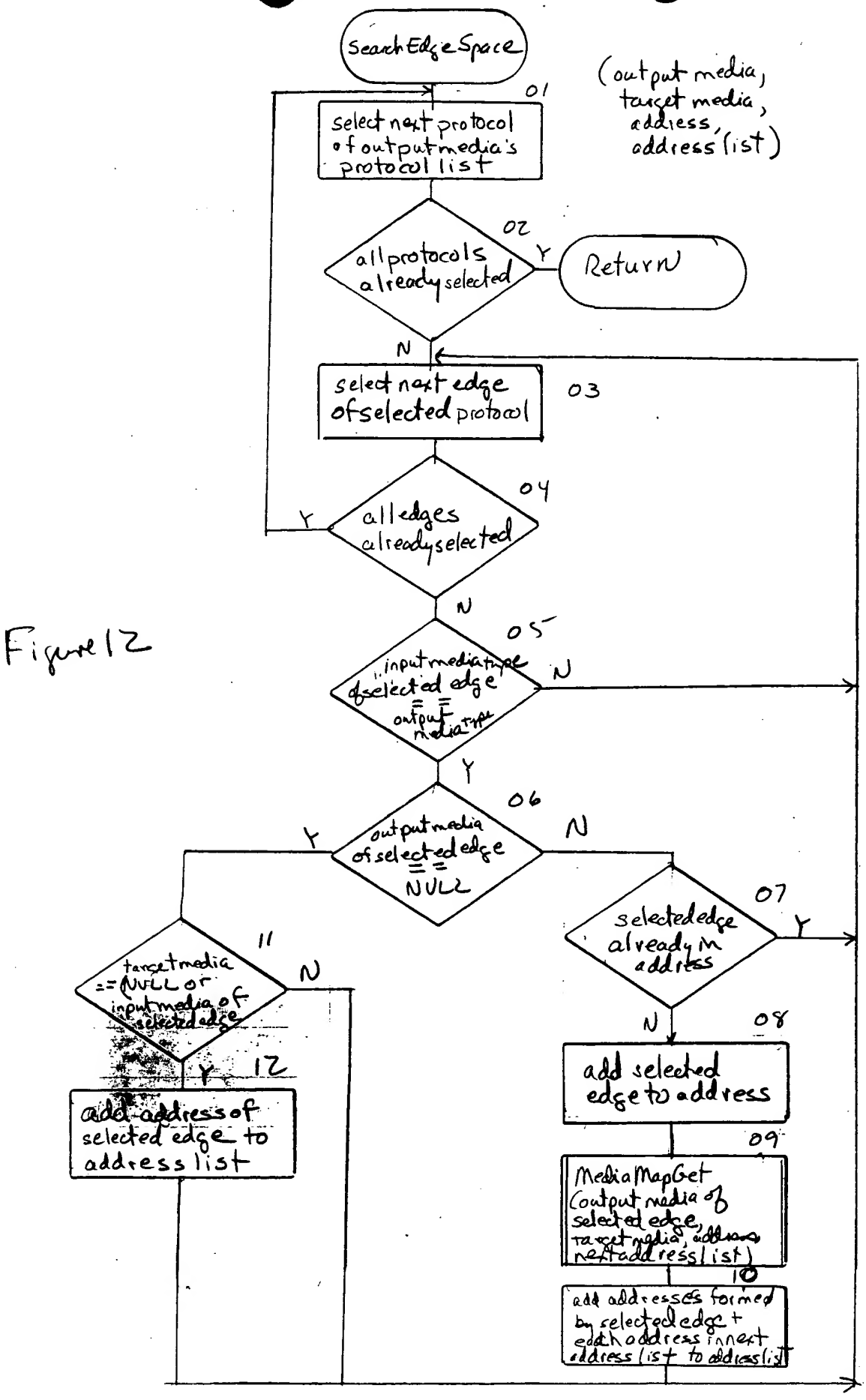

SearchEdgeSpace
(output media, target media, address, address list)
01
select next protocol of output media's protocol list
02
all protocols already selected
Y
Return
N
select next edge of selected protocol
03
04
all edges already selected
Y
N
05
input media type of selected edge == output media type
N
Y
06
output media of selected edge == NULL
Y
N
07
selected edge already in address
Y
N
08
add selected edge to address
09
MediaMapGet (output media of selected edge, target media, address, next address list)
10
add addresses formed by selected edge + each address in next address list to address list
11
target media == NULL or input media of selected edge
N
Y
12
add address of selected edge to address list

Figure 12

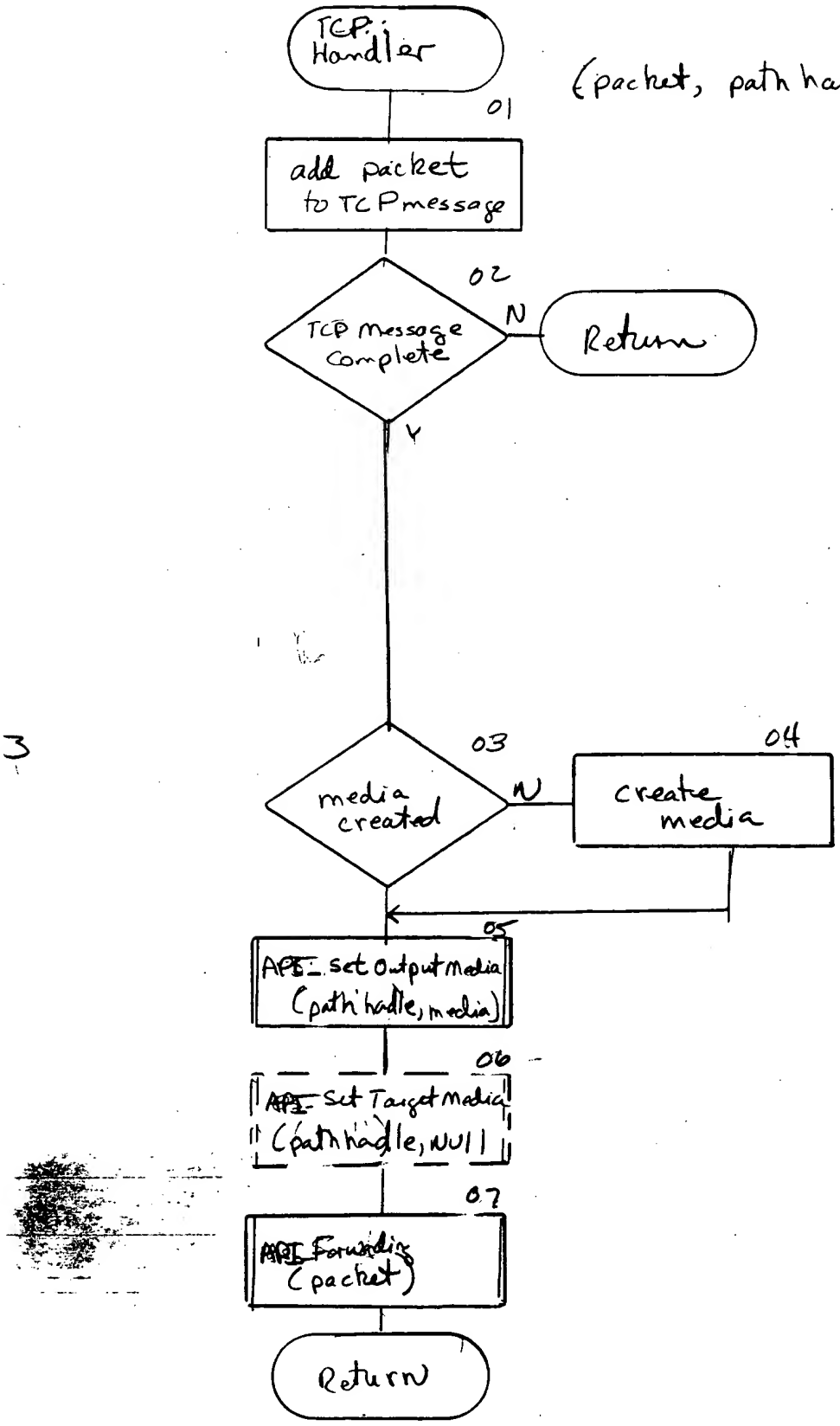

TCP Handler
(packet, path handle)
01
add packet to TCP message
02
TCP message complete
N
Return
Y
03
media created
N
04
create media
05
API - set Output Media (path handle, media)
06
API - set Target Media (path handle, NULL)
07
API - Forwarding (packet)
Return

Figure 13

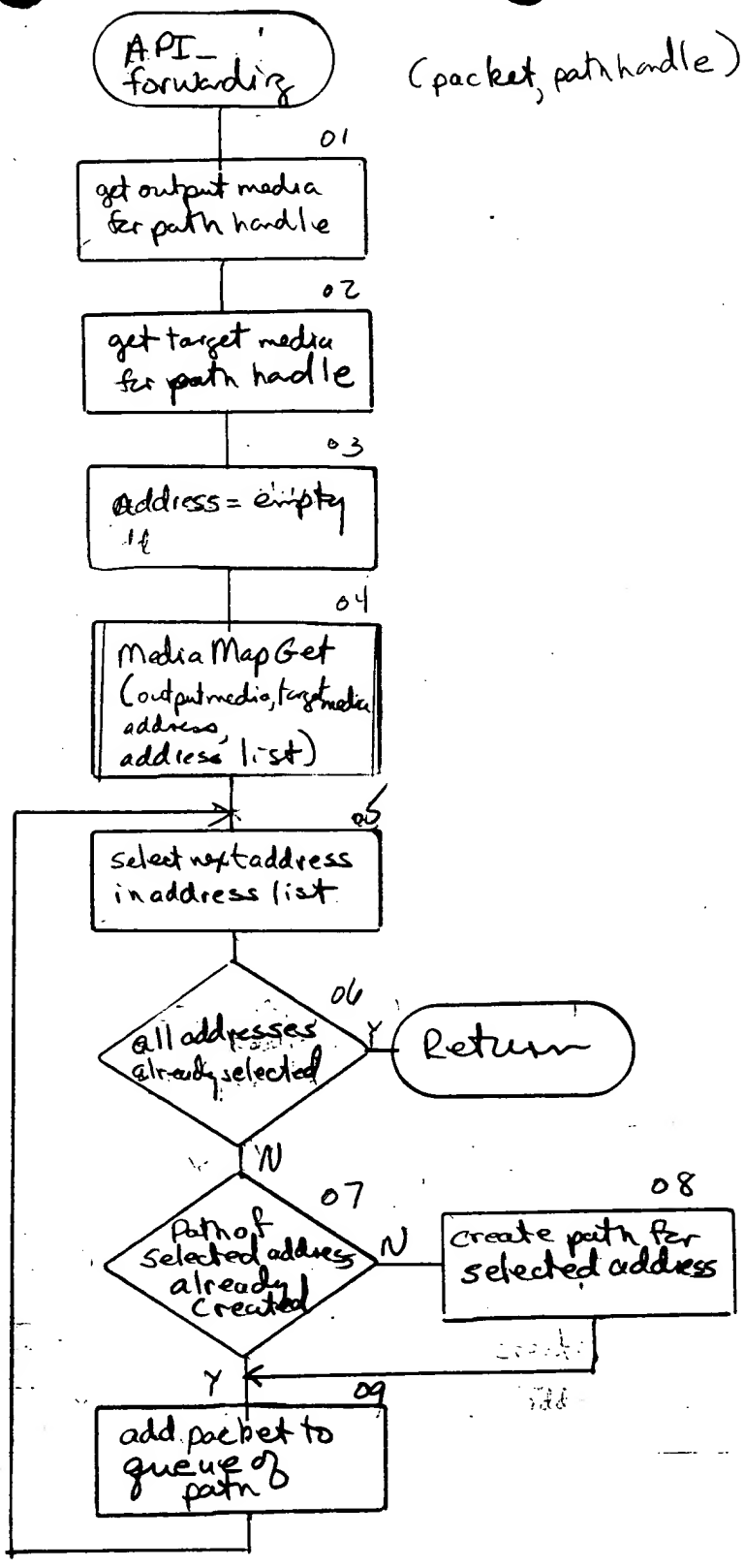

API_forwarding
(packet, pathhandle)
01
get output media for path handle
02
get target media for path handle
03
address = empty list
04
Media Map Get (output media, target media, address, address list)
05
select next address in address list
06
all addresses already selected
Y
Return
N
07
Path of selected address already created
N
08
create path for selected address
Y
09
add packet to queue of path

Figure 14

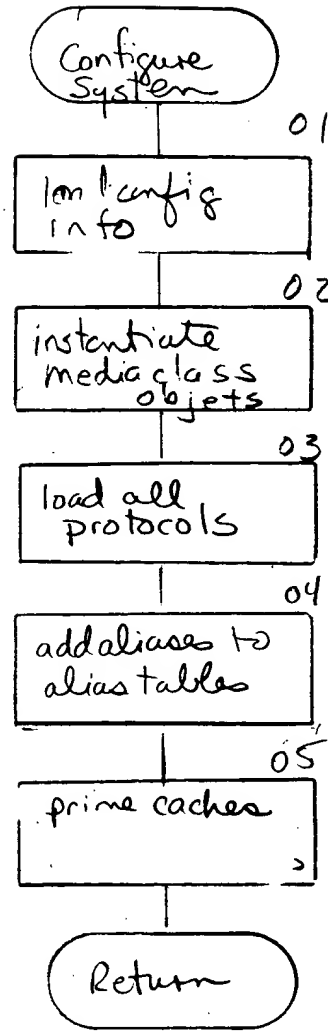

Configure System
01
Ion Config info
02
instantiate media class objects
03
load all protocols
04
add aliases to alias tables
05
prime caches
Return

Fig 15

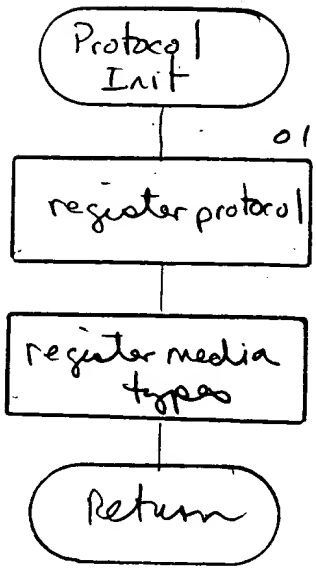

Protocol Init
01
register protocol
register media types
Return

Figure 16

Media Class 1

alias table 01

D1	
D2	D20
D5	D10
D9	
D10	
D12	D15
D13	D15
D15	
D19	
D20	

01a

Protocol list 02

P1
P2
P3
P4
P8

D1 Media Cache 05

D11	P1:1, P2:1, P3:2, P4:1
D15	P1:1, P5:2, P6:1, P7:1, P8:1

Media Class 2

alias table 03

D3	D4
D4	
D6	D7
D7	D11
D8	
D11	D7
D14	
D15	
D16	
D17	D19
D18	

Protocol List 04

P5
P6
P7
P9

Figure 17

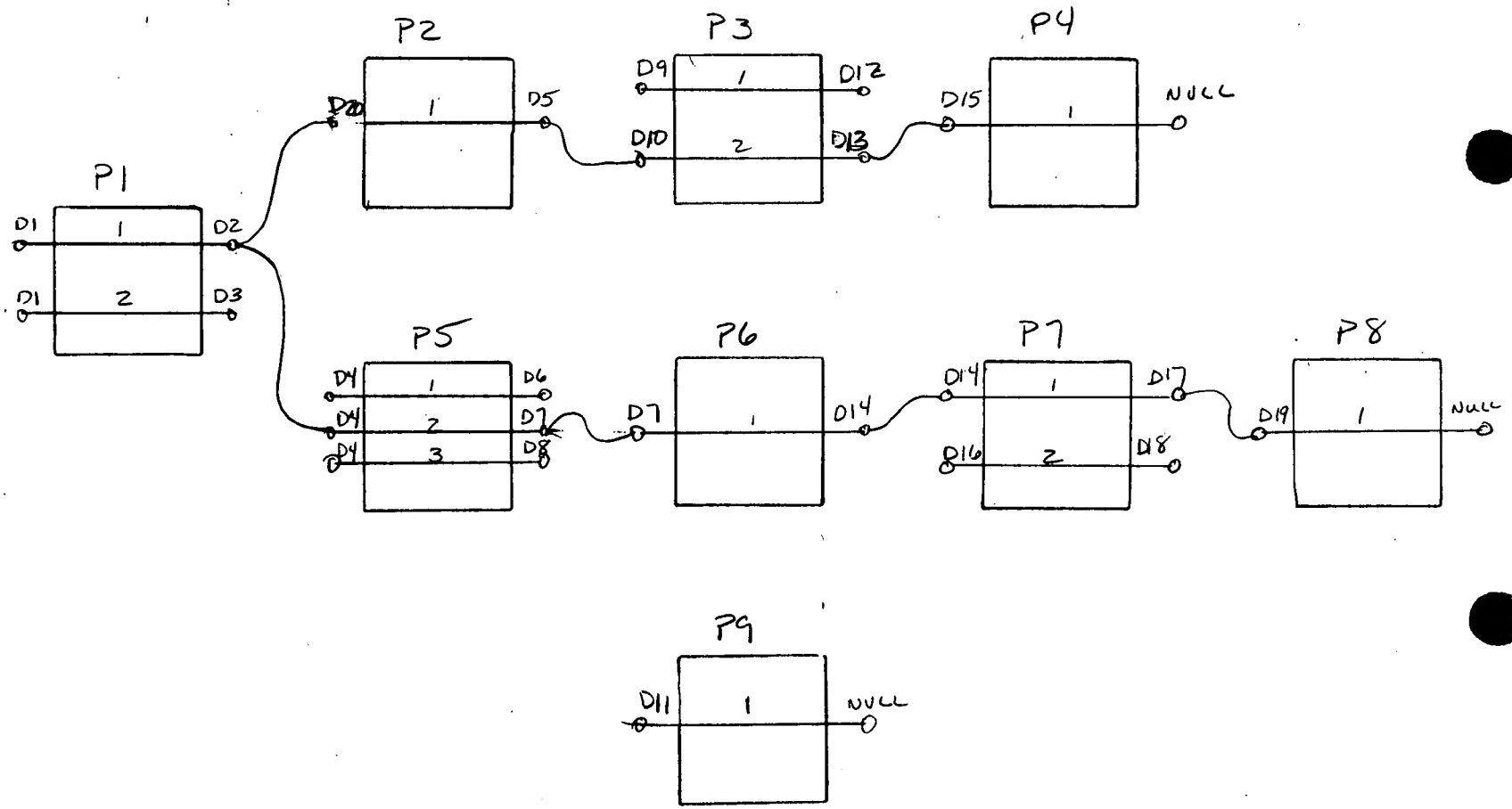

P1
D1
1
D2
D1
2
D3
P2
D4
1
D5
P3
D9
1
D12
D10
2
D13
P4
D15
1
NULL
P5
D4
1
D6
D4
2
D7
D4
3
D8
P6
D7
1
D14
P7
D14
1
D17
D16
2
D18
P8
D19
1
NULL
P9
D11
1
NULL

Figure 18

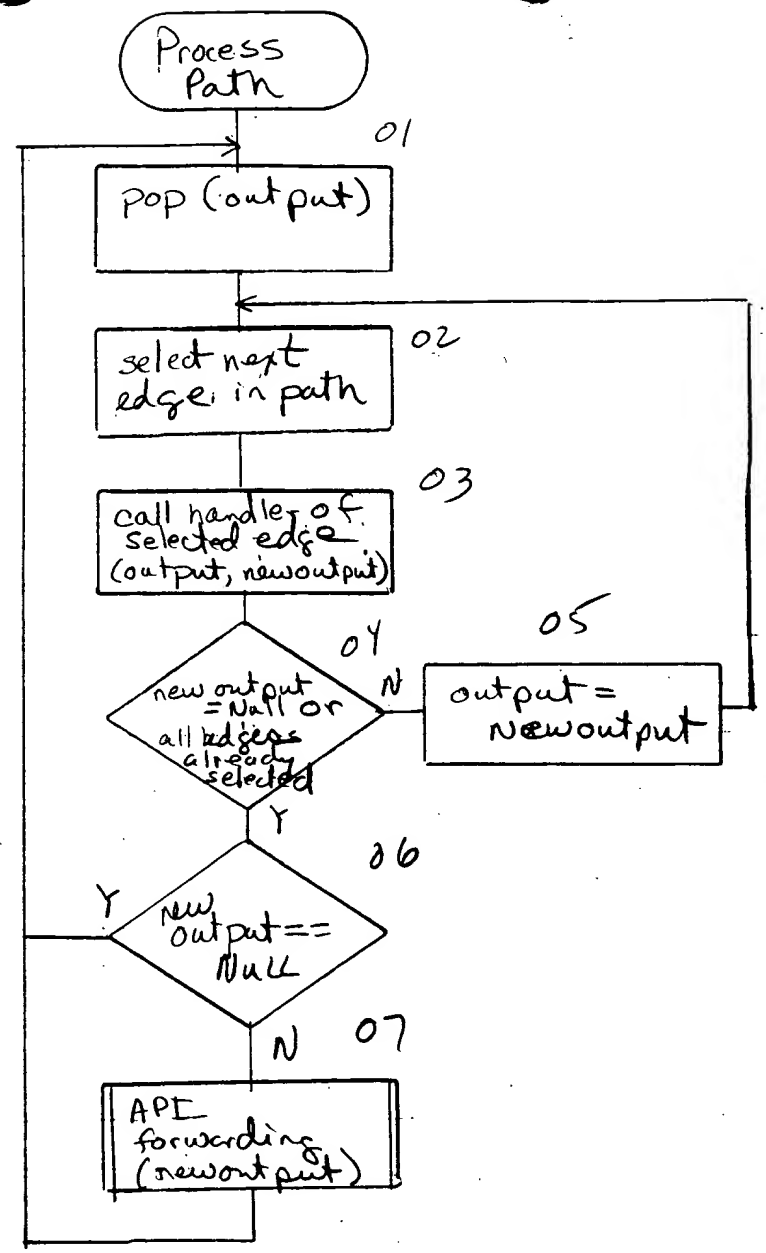

Process Path
01
pop (output)
02
select next edge in path
03
call handler of selected edge (output, newoutput)
04
new output = Null or all edges already selected
N
05
output = newoutput
Y
06
Y
new output == Null
N
07
API forwarding (newoutput)

Fig 19

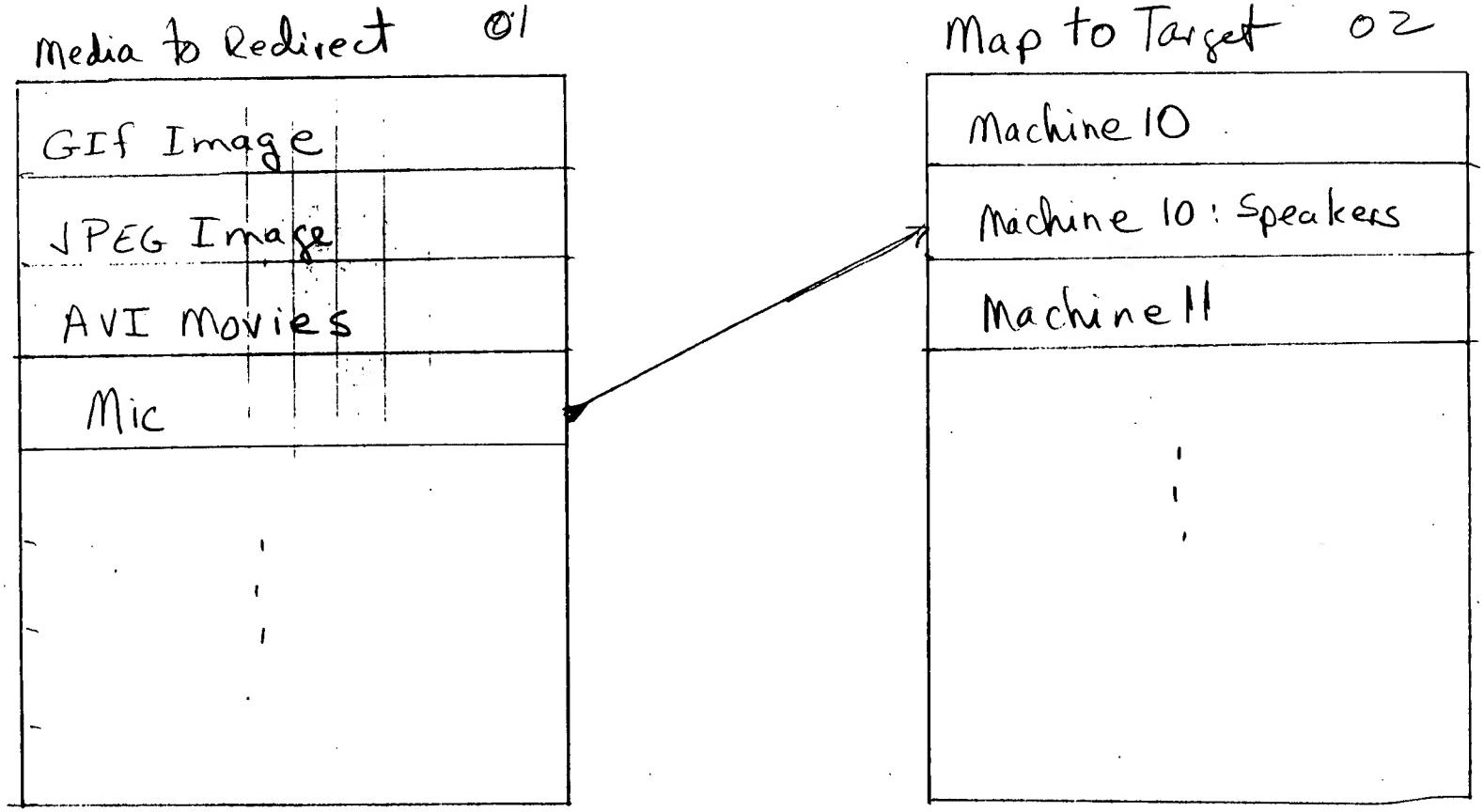
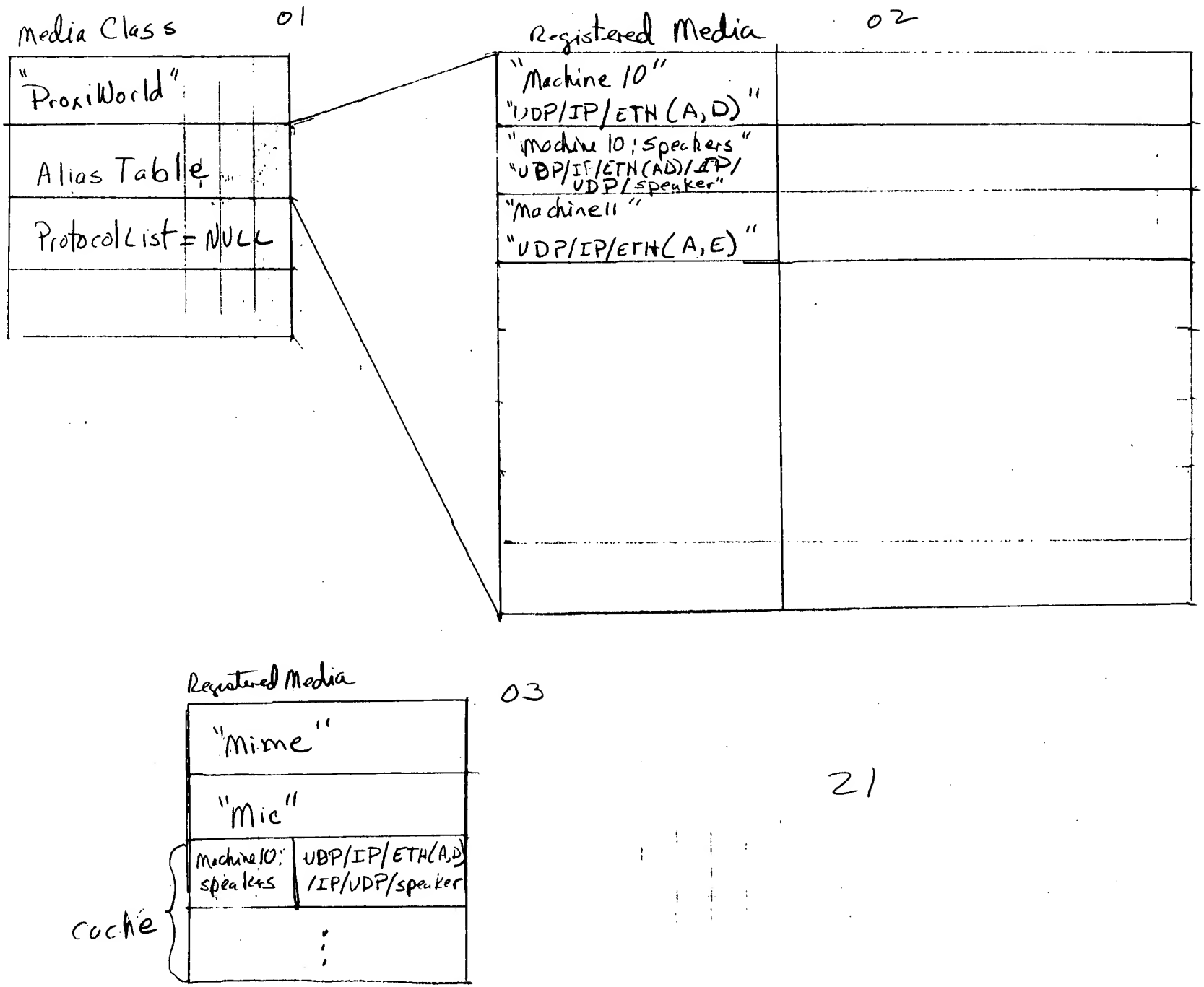
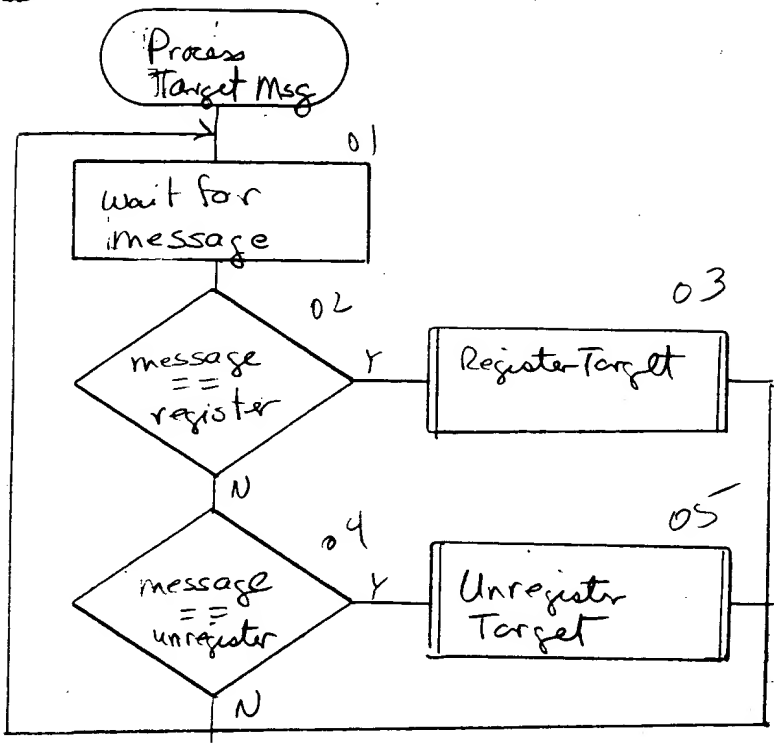
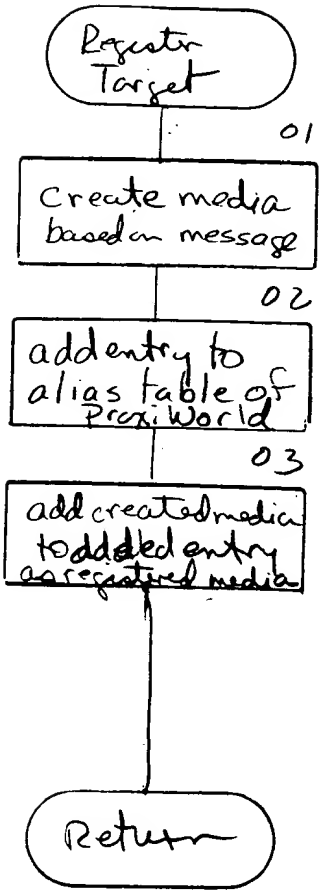

Media to Redirect
01
GIf Image
JPEG Image
AVI Movies
Mic
Map to Target
02
Machine 10
Machine 10 : Speakers
Machine 11

Fig 20

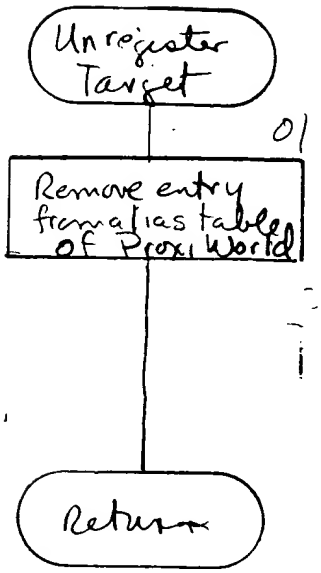

Media Class
01
"ProxiWorld"
Alias Table
ProtocolList = NULL
Registered Media
02
"Machine 10"
"UDP/IP/ETH (A,D)"
"machine 10; speakers"
"UDP/IP/ETH(AD)/IP/UDP/speaker"
"machine11"
"UDP/IP/ETH(A,E)"
Registered Media
03
"Mime"
"Mic"
machine10: speakers
UDP/IP/ETH(A,D) /IP/UDP/speaker
cache


Process Target Msg
01
wait for message
02
message == register
Y
03
Register Target
N
04
message == unregister
Y
05
Unregister Target
N

22


Register Target
(message)
01
Create media based on message
02
add entry to alias table of ProxiWorld
03
add created media to added entry as registered media
Return

23


Unregister Target
(message)
01
Remove entry from alias table of Proxi World
Return

24